

AD-A269 592



University  
of Southern  
California



12

## Granularity in Multi-Method Planning

Soowon Lee and Paul S. Rosenbloom  
USC/ Information Sciences Institute  
4676 Admiralty Way  
Marina del Rey, California 90292

May 1993  
ISI/RR-93-351

DTIC  
ELECTE  
SEP 21 1993  
S B D

DISTRIBUTION STATEMENT A  
Approved for public release  
Distribution Unlimited

INFORMATION  
SCIENCES  
INSTITUTE



310/822-1511  
4676 Admiralty Way/Marina del Rey/California 90292-6695

93

648

# **Granularity in Multi-Method Planning**

Soowon Lee and Paul S. Rosenbloom  
USC/ Information Sciences Institute  
4676 Admiralty Way  
Marina del Rey, California 90292

May 1993  
ISI/RR-93-351

**93-21768**



**REPORT DOCUMENTATION PAGE**FORM APPROVED  
OMB NO. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE May 1993	3. REPORT TYPE AND DATES COVERED Research Report	
4. TITLE AND SUBTITLE Granularity in Multi-Method Planning			5. FUNDING NUMBERS N00014-89-K-0155	
6. AUTHOR(S) Soowon Lee and Paul S. Rosenbloom				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) USC INFORMATION SCIENCES INSTITUTE 4676 ADMIRALTY WAY MARINA DEL REY, CA 90292-6695			8. PERFORMING ORGANIZATION REPORT NUMBER RR-351	
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES) ARPA 3701 Fairfax Drive Arlington, VA 22203			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12A. DISTRIBUTION/AVAILABILITY STATEMENT UNCLASSIFIED/UNLIMITED			12B. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Multi-method planning is an approach to using a set of different planning methods to simultaneously achieve planner completeness, planning time efficiency, and plan length reduction. Although it has been shown that coordinating a set of methods in a course-grained, problem-by-problem manner has the potential for approaching this ideal, such an approach can waste a significant amount of time in trying methods that ultimately prove inadequate. This paper investigates an approach to reducing this wasted effort by refining the granularity at which methods are switched. The experimental results show that the fine-grained approach can improve the planning time significantly compared with coarse-grained and single-method approaches.				
14. SUBJECT TERMS Multi-method planning, Coarse-grained planners, Fine-grained planners, Strong monotonicity, Planning bias			15. NUMBER OF PAGES 7	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UNLIMITED	

## GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to stay within the lines to meet optical scanning requirements.

**Block 1. Agency Use Only (Leave blank).**

**Block 2. Report Date.** Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

**Block 3. Type of Report and Dates Covered.** State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

**Block 4. Title and Subtitle.** A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

**Block 5. Funding Numbers.** To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

**Block 6. Author(s).** Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

**Block 7. Performing Organization Name(s) and Address(es).** Self-explanatory.

**Block 8. Performing Organization Report Number.** Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

**Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es).** Self-explanatory

**Block 10. Sponsoring/Monitoring Agency Report Number.** (If known)

**Block 11. Supplementary Notes.** Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of ...; To be published in... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

**Block 12a. Distribution/Availability Statement.** Describe public availability or limitations. Cite any available to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."  
DOE - See authorities.  
NASA - See Handbook NHB 2200.2.  
NTIS - Leave blank.

**Block 12b. Distribution Code.**

DOD - Leave blank.  
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.  
NASA - Leave blank.  
NTIS - Leave blank.

**Block 13. Abstract.** Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

**Block 14. Subject Terms.** Keywords or phrases identifying major subjects in the report.

**Block 15. Number of Pages.** Enter the total number of pages.

**Block 16. Price Code.** Enter appropriate price code (NTIS only).

**Blocks 17.-19. Security Classifications.** Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

**Block 20. Limitation of Abstract.** This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

# Granularity in Multi-Method Planning\*

Soowon Lee and Paul S. Rosenbloom

Information Sciences Institute and Computer Science Department

University of Southern California

4676 Admiralty Way

Marina del Rey, CA 90292

{swlee,rosenbloom}@isi.edu

(To appear in *Proceedings of the Eleventh National Conference on AI*, 1993.)

## Abstract

Multi-method planning is an approach to using a set of different planning methods to simultaneously achieve planner completeness, planning time efficiency, and plan length reduction. Although it has been shown that coordinating a set of methods in a coarse-grained, problem-by-problem manner has the potential for approaching this ideal, such an approach can waste a significant amount of time in trying methods that ultimately prove inadequate. This paper investigates an approach to reducing this wasted effort by refining the granularity at which methods are switched. The experimental results show that the fine-grained approach can improve the planning time significantly compared with coarse-grained and single-method approaches.

## Introduction

The ability to find a low execution-cost plan efficiently over a wide domain of applicability is the core of domain-independent planning systems. The key issue here is how to construct a single planning method, or how to coordinate a set of different planning methods, that has sufficient scope and efficiency. Our approach to this issue begins with the observation that no single method will satisfy both sufficiency and efficiency, with the implication therefore that a coordinated set of planning methods will be needed.

We have constructed a system that can utilize six different planning methods, based on the notion of *bias* in planning. A planning bias is any constraint over the space of plans considered that determines which portion of the entire plan space can be the output of the planning.<sup>1</sup> The six planning methods used vary along two independent bias dimensions: goal-protection and

goal-flexibility. The goal-protection dimension determines whether or not a protection bias is used, that eliminates plans in which an operator undoes an initial goal conjunct that is either true a priori or established by an earlier operator in the sequence. The goal-flexibility dimension determines the degree of flexibility the planner has in using new subgoals. Two biases, directness and linearity, are used along this dimension. Directness eliminates plans in which operators are used to achieve preconditions of other operators, rather than just top-level goal conjuncts. Linearity eliminates plans in which operators for different goal conjuncts are interleaved. The  $3 \times 2$  methods arise from the cross-product of these two dimensions: (directness, linearity, or nonlinearity)  $\times$  (protection, or no-protection).<sup>2</sup>

These single-method planners are implemented in the context of the Soar architecture (Laird, Newell, & Rosenbloom, 1987). Plans in Soar are represented as sets of control rules that jointly specify which operators should be executed at each point in time (Rosenbloom, Lee, & Unruh, 1990). Planning time for these methods is measured in terms of *decisions*, the basic behavioral cycle in Soar. This measure is not quite identical to the more traditional measure of number of planning operators executed, but should still correlate with it relatively closely.

The six implemented methods have previously been compared empirically in terms of planner completeness, planning time, and plan length over a test set of 100 randomly generated 3- and 4-conjunct problems in the blocks-world domain. The predominant result obtained so far from the experiments with these methods is that planning time and plan length are both inversely correlated with the applicability of the plan-

does not rule out a search strategy that incrementally specifies an element of the plan space by refining a partially-ordered plan structure.

<sup>2</sup>The term "nonlinearity" in this context implies that it is allowable to interleave operators in service of different goal conjuncts. It does not necessarily mean that either partial-order or least-commitment planning are being used.

\*This work was sponsored by the Defense Advanced Research Projects Agency (DOD) and the Office of Naval Research under contract number N00014-89-K-0155.

<sup>1</sup>The specification here assumes that the plan space contains only totally-ordered sequences of operators, but it

Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

ning method; that is, the more restricted the method, the less time it takes to solve the problems that it can solve, and the shorter are the plans generated. The most restricted method (the method with directness and protection) could solve 68 of them, in an average of 16.3 decisions each, producing plans containing an average of 1.8 operators (Lee & Rosenbloom, 1992). The least restricted method (nonlinear planning without goal protection) could solve all 100 problems; however, planning time and plan length averaged over the same 68 problems solvable by the most restricted method were considerably worse — an average of 39.0 decisions to produce plans containing on average 3.3 operators.

This trade-off between completeness and efficiency implies that the planning system would be best served if it could always opt for the most restricted method adequate for its current situation. In a first step towards this ideal, we have begun exploring multi-method planners that start by trying highly restricted methods, and then successively relax the restrictions until a method is found that is sufficient for the problem. The intuition behind this is based on iterative deepening (Korf, 1985) — if the proportion of problems solvable at a particular level of restriction is large enough, and the ratio of costs for successive levels is large enough, there should be a net gain. Over the set of 100 blocks-world problems, this has yielded broadly applicable multi-method planners (actually, complete for the blocks-world) that on average generate shorter plans than are produced by corresponding (complete) single-method planners, with marginally lower planning times (from 39.9 to 52.5 decisions for single-method planners versus from 33.4 to 42.2 decisions for multi-method planners).

However these results do not necessarily mean that, for all situations, there exists a multi-method planner which outperforms the most efficient single-method planner. In fact, the performance of these planners depends on the biases used in the multi-method planners and the problem set used in the experiments. For example, if the problems are so complex that most of the problems are solvable only by the least restricted method, the performance loss by trying inappropriate earlier methods in multi-method planners might be relatively considerable. On the other hand, if the problems are so trivial that it takes only a few decisions for the least restricted method to solve the problems, the slight performance gain by using more restricted methods in multi-method planners might be overridden by the complexity of the meta-level processing required to coordinate the sequence of primitive planners.

These results suggest that multi-method planning is a promising approach, but that further work is necessary to establish whether robust gains are possible over a wide range of domains. The work reported here is

one step in this direction, in which we investigate reducing the wasted effort in multi-method planners by refining the granularity at which the individual planning methods can be switched. This approach has been implemented, and initial experiments in two domains show significant gains in planning time with respect to both single-method and the earlier, coarser-grained, multi-method planners.

## Fine-grained Multi-method Planners

The approach to multi-method planning described so far starts with a restricted method and switches to a less restricted method whenever the current method fails. This switch is always made on a problem-by-problem basis. However, this is not the only granularity at which methods could be switched. The family of multi-method planning systems can be viewed on a granularity spectrum. While in coarse-grained multi-method planners, methods are switched for a whole problem when no solution can be found for the problem within the current method, in fine-grained multi-method planners, methods can be switched at any point during a problem at which a new set of subgoals is formulated, and the switch only occurs for that set of subgoals (and not for the entire problem). At this finer level of granularity it is conceivable that the planner could use a highly-restricted and efficient method over much of a problem, but fall back on a nonlinear method without protection for those critical subregions where there are tricky interactions.

With this flexibility of method switching, fine-grained multi-method planning can potentially outperform both coarse-grained multi-method planning and single-method planning. Compared with coarse-grained multi-method planning, it can save the effort of backtracking when the current method can not find a solution or the current partial plan violates the biases used in the current method. Moreover, it can save the extra effort of using a less restricted method on later parts of the problem, just because one early part requires it. As compared with single-method planning, a fine-grained multi-method planner can utilize biases which would cause incompleteness in a single-method planner — such as directness or protection in the blocks-world domain — while still remaining complete. The result is that a fine-grained multi-method planner can potentially be more efficient than a single-method planner that has the same coverage of solvable problems.

One way to construct an efficient multi-method planner is to order the single method planners according to increasing coverage and decreasing efficiency, an approach called *monotonic multi-method planning*. In this paper, we focus on a special type of monotonic multi-method planner, called a *strongly monotonic multi-method planner*, which is based on the de-

Planner	Decisions			Plan length		
	$A_1$	$A_2$	$A_5$	$A_1$	$A_2$	$A_5$
$M_1$ (directness, protection)	12.50	-	-	1.56	-	-
$M_2$ (linearity, protection)	13.00	18.90	-	1.56	2.32	-
$M_3$ (protection)	13.21	26.91	-	1.62	2.49	-
$M_4$ (directness)	14.48	-	-	1.71	-	-
$M_5$ (linearity)	14.81	24.47	24.84	2.10	3.22	3.34
$M_6$	16.23	40.85	40.96	2.02	3.17	3.37

Table 1: The performance of the six single-method planners for the three problem sets defined by the scopes of the planners.

liberate selection and relaxation of effective biases. In the next section, we provide a formal definition of a monotonic multi-method planner, and define a criterion for selecting effective biases from experiments with single-method planners.

### Selecting Effective Biases

Let  $M_{k_i}$  ( $k_i \in \{1, \dots, 6\}$ ) be a single-method planner, as defined in Section 1. A fine-grained multi-method planner that consists of a sequence of  $n$  different single-method planners is denoted as  $M_{k_1-k_2-\dots-k_n}$ , and the corresponding coarse-grained multi-method planner is denoted as  $M_{k_1} \rightarrow M_{k_2} \rightarrow \dots \rightarrow M_{k_n}$ . Let  $A$  be a sample set of problems, and let  $A_k \subseteq A$  be the subset of  $A$  which are solvable in principle by  $M_{k_i}$ . The functions  $s(M_{k_i}, A_s)$  and  $l(M_{k_i}, A_s)$  represent respectively the average cost that  $M_{k_i}$  requires to succeed and the average length of plans generated by  $M_{k_i}$ , for the problems in  $A_s \subseteq A_{k_i}$ . Let  $M_{k_0}$  be a null planner which cannot solve any problems; that is,  $A_{k_0} = \emptyset$ .

A multi-method planner which consists of  $M_{k_1}, M_{k_2}, \dots, M_{k_n}$  is called *monotonic* if the following three conditions hold for each pair of  $M_{k_{i-1}}$  and  $M_{k_i}$ , for  $2 \leq i \leq n$ : (1)  $A_{k_{i-1}} \subseteq A_{k_i}$ , (2)  $s(M_{k_{i-1}}, A_{k_{i-1}}) \leq s(M_{k_i}, A_{k_{i-1}})$ , for  $j \leq i$ , and (3)  $l(M_{k_{i-1}}, A_{k_{i-1}}) \leq l(M_{k_i}, A_{k_{i-1}})$ , for  $j \leq i$ .<sup>3</sup> The straightforward way to build monotonic multi-method planners is to run each of the individual methods on a set of training problems, and then from the resulting data to generate all method sequences for which monotonicity holds. The approach we have taken here is to generate only a subset of this full set; in particular, we have focused only on multi-method planners in which later methods embody subsets of the biases incorporated into earlier methods, and in which the biases themselves are all *positive*.

Let  $B_{k_i}$  be the set of biases used in  $M_{k_i}$ . A bias  $b$  is called *positive* in a problem set  $A$  and a method

set  $\{M_{k_i}\}$ , if for each pair of  $M_{k_x}$  and  $M_{k_y}$  in  $\{M_{k_i}\}$  such that  $B_{k_x} = B_{k_y} + \{b\}$ ,  $s(M_{k_x}, A_{k_j}) \leq s(M_{k_y}, A_{k_j})$  and  $l(M_{k_x}, A_{k_j}) \leq l(M_{k_y}, A_{k_j})$ , for every  $j \leq x$ . A multi-method planner which consists of  $M_{k_1}, M_{k_2}, \dots, M_{k_n}$  is called *strongly monotonic* if  $B_{k_{i-1}} \supset B_{k_i}$ , for  $2 \leq i \leq n$ , and  $B_{k_{i-1}} - B_{k_i}$  consists of positive biases only, for  $2 \leq i \leq n$ . From this definition, if a multi-method planner is strongly monotonic, it is monotonic, while the reverse is not necessarily true.

To generate a strongly monotonic multi-method planner, it is necessary to determine which biases are positive in the domain. Table 1 illustrates the average number of decisions,  $s(M_{k_i}, A_{k_j})$ , and average plan lengths,  $l(M_{k_i}, A_{k_j})$  for the six single-method planners and the problem sets defined by the scope of these planners over a training set of 30 randomly generated 3- and 4-conjunct problems in the blocks-world domain. In this domain,  $A_4$  is the same as  $A_1$  because if a problem is not solvable with protection, it also is not solvable with directness.  $A_5$  is the same as  $A_6$  because both  $M_5$  and  $M_6$  are complete in this domain, though  $M_5$  may not be able to generate an optimal solution.  $A_2$  and  $A_3$  are different sets in principle, because problems such as Sussman's anomaly cannot be solved by a linear planner with protection ( $M_2$ ) but can be by a nonlinear planner with protection ( $M_3$ ). However, among the 30 problems, these "anomaly" problems did not occur, yielding  $A_2 = A_3$  for this set of problems. The results imply that directness and protection are positive in this domain, while linearity is not, since  $l(M_5, A_1) > l(M_6, A_1)$  and  $l(M_5, A_2) > l(M_6, A_2)$ . If we use linearity as an independent bias — so that one set of multi-method planners is generated using it and one set without it — and vary directness and protection within the individual multi-method planners, we get a set of 10 strongly monotonic multi-method planners (four three-method planners and six two-method planners).

<sup>3</sup>This is a slight redefinition of monotonicity from (Lee & Rosenbloom, 1992) with a minor correction.

Planner	Decisions				Plan length			
	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>5</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>5</sub>
M <sub>5</sub>	22.21	29.41	29.48	29.22	3.00	3.78	3.83	3.82
M <sub>6</sub>	33.40	47.12	48.06	47.93	2.90	3.88	4.07	4.14
Average				38.58				3.98
M <sub>1</sub> →M <sub>2</sub> →M <sub>5</sub>	13.26	24.69	25.07	26.13	1.82	2.48	2.54	2.58
M <sub>1</sub> →M <sub>3</sub> →M <sub>6</sub>	13.26	26.34	26.55	28.91	1.82	2.52	2.54	2.59
M <sub>1</sub> →M <sub>4</sub> →M <sub>5</sub>	13.26	26.16	26.41	26.79	1.82	2.85	2.92	2.94
M <sub>1</sub> →M <sub>4</sub> →M <sub>6</sub>	13.26	36.78	37.40	37.30	1.82	2.91	2.99	3.02
M <sub>1</sub> →M <sub>5</sub>	13.26	25.68	25.86	26.04	1.82	2.96	3.02	3.03
M <sub>1</sub> →M <sub>6</sub>	13.26	31.54	31.85	31.77	1.82	2.89	2.94	2.97
M <sub>2</sub> →M <sub>5</sub>	19.54	27.89	28.18	29.34	1.85	2.43	2.49	2.58
M <sub>3</sub> →M <sub>6</sub>	21.22	28.46	28.41	30.67	2.00	2.52	2.52	2.57
M <sub>4</sub> →M <sub>5</sub>	16.85	27.81	27.95	28.38	1.82	2.83	2.88	2.93
M <sub>4</sub> →M <sub>6</sub>	16.85	33.33	33.59	34.47	1.82	2.83	2.85	2.95
Average				29.98				2.82
M <sub>1</sub> -2-5	8.63	12.87	13.00	13.01	1.82	2.80	2.84	2.90
M <sub>1</sub> -3-6	8.63	13.38	13.43	13.56	1.82	2.53	2.53	2.59
M <sub>1</sub> -4-5	8.63	13.19	13.29	13.25	1.82	3.25	3.32	3.34
M <sub>1</sub> -4-6	8.63	13.48	13.73	13.63	1.82	2.87	2.96	2.97
M <sub>1</sub> -5	8.63	12.21	12.36	12.51	1.82	2.63	2.73	2.81
M <sub>1</sub> -6	8.63	13.22	13.27	13.23	1.82	2.68	2.69	2.73
M <sub>2</sub> -5	19.19	23.75	23.76	23.80	2.56	3.07	3.11	3.16
M <sub>3</sub> -6	16.62	23.45	23.56	24.22	2.03	2.56	2.57	2.71
M <sub>4</sub> -5	13.57	17.24	17.30	17.38	2.44	3.71	3.77	3.77
M <sub>4</sub> -6	14.10	19.28	19.58	19.83	2.41	3.33	3.43	3.46
Average				16.44				3.04

Table 2: Single-method and coarse-grained multi-method vs. fine-grained multi-method planning in the blocks-world domain.

## Experimental Results

Table 2 compares the strongly monotonic fine-grained multi-method planners with the corresponding coarse-grained multi-method planners and (complete) single-method planners over a test set of 100 randomly generated 3- and 4-conjunct blocks-world problems (this test set is disjoint from the 30-problem training set used in developing the multi-method planners). Z-tests on this data reveal that fine-grained multi-method planners take significantly less planning time than both single-method planners ( $z=5.35$ ,  $p<.01$ ) and coarse-grained multi-method planners ( $z=6.72$ ,  $p<.01$ ). This likely stems from fine-grained multi-method planners preferring to search within the more efficient spaces defined by the biases — thus tending to outperform single-method planners — but being able to recover from bias failure without throwing away everything already done for a problem (thus tending to outperform coarse-grained multi-method planners).

Fine-grained multi-method planners also generate significantly shorter plans than single-method planners

( $z=3.42$ ,  $p<.01$ ). They generate slightly longer plans than coarse-grained multi-method planners; however, no significance is found at a 5% level ( $z=1.77$ ). These results likely arise because, whenever possible, both types of multi-method planners use the more restrictive methods that yield shorter plan lengths, while there may be little difference between the methods that ultimately succeed for the two types of multi-method planners.

Table 3 illustrates the performance of these three types of planners over a test set of 100 randomly generated 5-conjunct problems in the machine shop scheduling domain (Minton, 1988). In this domain, no precondition subgoals are required because there is no operator which achieves any of the unmet preconditions. Thus both directness and linearity are irrelevant. However, there are strong interactions among the operators, so protection violations are still relevant. In consequence, the entire table of six planners reduces to only two distinct planners for this domain: with or without protection.



Planner	Decisions		Plan length	
	A <sub>1</sub>	A <sub>4</sub>	A <sub>1</sub>	A <sub>4</sub>
$M_4, M_5, M_6$	31.47	33.97	4.13	4.47
$M_1 \rightarrow M_4, M_2 \rightarrow M_5, M_3 \rightarrow M_6$	26.17	35.91	2.43	3.58
$M_1 \rightarrow 4, M_2 \rightarrow 5, M_3 \rightarrow 6$	18.71	19.07	2.87	3.29

Table 3: Single-method and coarse-grained multi-method vs. fine-grained multi-method planning in the scheduling domain.

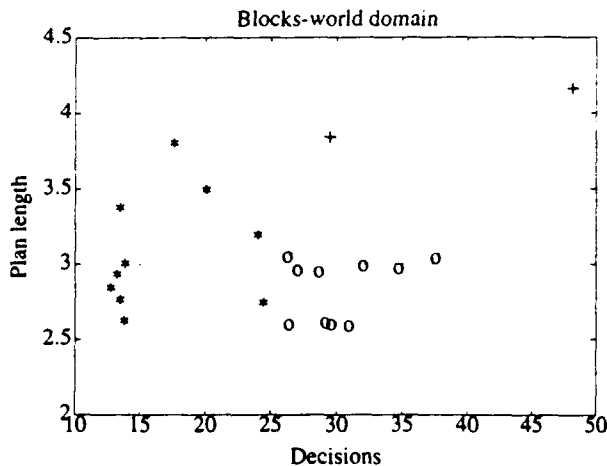


Figure 1: Performance of single-method planners (+), coarse-grained multi-method planners (o), and fine-grained multi-method planners (\*) in the blocks-world domain.

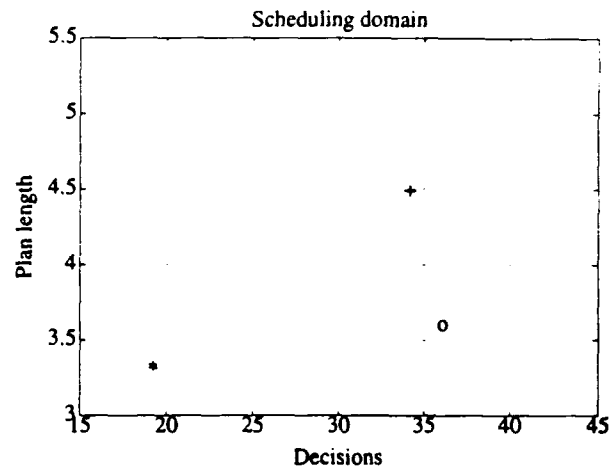


Figure 2: Performance of single-method planners (+), coarse-grained multi-method planners (o), and fine-grained multi-method planners (\*) in the scheduling domain.

As with the blocks-world domain, the z-tests in the scheduling domain indicate that fine-grained planners dominate both single-method planners ( $z=10.91$ ,  $p<.01$ ) and coarse-grained planners ( $z=8.95$ ,  $p<.01$ ) in terms of planning time. Fine-grained planners also generate significantly shorter plans than do the single-method planners ( $z=6.49$ ,  $p<.01$ ). They generate slightly shorter plans than coarse-grained multi-method planners; however, no significance is found at a 5% level ( $z=1.28$ ).

Figures 1 and 2 plot the average number of decisions versus the average plan lengths for the data in Tables 2 and 3. These figures graphically illustrate how the coarse-grained approach primarily reduces plan length in comparison to the single-method approach, and how the fine-grained approach primarily improves efficiency in comparison to the coarse-grained approach.

## Related Work

The basic approach of bias relaxation in multi-method planning is similar to the shift of bias for inductive con-

cept learning (Russell & Grosz, 1987; Utgoff, 1986). In the planning literature, this approach is closely related to an *ordering modification* which is a control strategy to prefer exploring some plans before others (Gratch & DeJong, 1990). Bhatnagar & Mostow (1990) described a relaxation mechanism for over-general censors in FAILSAFE-2. However, there are a number of differences, such as the type of constraints used, the granularity at which censors are relaxed, and the way censors are relaxed. SteppingStone (Ruby & Kibler, 1991) tries constrained search first, and moves on to unconstrained search, if the constrained search reaches an impasse (within the boundary of ordered subgoals) and the knowledge stored in memory cannot resolve the impasse.

This approach is also related to the traditional partial-order planning, where heuristics are used to guide the search over the space of partially ordered plans without violating planner completeness (McAllester & Rosenblitt, 1991; Barrett & Weld, 1993; Chapman, 1987). For example, using directness in fine-grained multi-method planners is similar to preferring

the nodes which reduce the size of the set of open conditions when a new step is added. Relaxing bias in fine-grained multi-method planners only when it is necessary is similar to the least-commitment approach which adds ordering constraints only if a threat to a causal link is detected.

## Conclusion

In this paper, we have provided a way to select a set of positive biases for multi-method planning and investigated the effect of refining the granularity at which individual planning methods could be switched. The experimental results obtained so far in the blocks-world and machine-shop-scheduling domains imply that (1) fine-grained multi-method planners can be significantly more efficient than single-method planners in terms of planning time and plan length, and (2) fine-grained multi-method planners can be significantly more efficient than coarse-grained multi-method planners in terms of planning time.

The bias selection approach used here is based on preprocessing a set of training examples in order to develop fixed sequences of biases (and methods). A more dynamic, run-time approach would be to learn, while doing, which biases (and methods) to use for which classes of problems. If such learned information can transfer to the later problems, much of the effort wasted in trying inappropriate methods, as well as the effort for preprocessing, may be reduced (as demonstrated in (Rosenbloom, Lee, & Unruh, 1993)).

Another way to enhance the multi-method planning framework would be to extend the set of biases available to include ones that limit the size of the goal hierarchy (to reduce the search space), limit the length of plans generated (to shorten execution time), and lead to learning more effective rules (to increase transfer) (Etzioni, 1990). Investigations of these topics are in progress.

## References

Barrett, A., & Weld, D. S. (1993). Partial-order planning: Evaluating possible efficiency gains. *Artificial Intelligence*. To appear.

Bhatnagar, N., & Mostow, J. (1990). Adaptive search by explanation-based learning of heuristic censors. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 895-901). Boston, MA: AAAI Press.

Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence*, 32, 333-377.

Etzioni, O. (1990). Why Prodigy/EBL works. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 916-922). Boston, MA: AAAI Press.

Gratch, J. M., & DeJong, G. F. (1990). A framework for evaluating search control strategies. *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control* (pp. 337-347). San Diego, CA: Morgan Kaufmann.

Korf, R. E. (1985). Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27, 97-109.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33, 1-64.

Lee, S., & Rosenbloom, P. S. (1992). Creating and coordinating multiple planning methods. *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence* (pp. 89-95).

McAllester, D., & Rosenblitt, D. (1991) Systematic Nonlinear Planning. *Proceedings of the Ninth National Conference on Artificial Intelligence* (pp. 634-639). Anaheim, CA: AAAI Press.

Minton, S. (1988). Learning effective search control knowledge: An explanation-based approach. (Ph.D. Thesis). Department of Computer Science, Carnegie-Mellon University.

Rosenbloom, P. S., Lee, S., & Unruh, A. (1990). Responding to impasses in memory-driven behavior: A framework for planning. *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling, and Control* (pp. 181-191). San Diego, CA: Morgan Kaufmann.

Rosenbloom, P. S., Lee, S., & Unruh, A. (1993). Bias in planning and explanation-based learning. S. Chipman & A. L. Meyrowitz (Eds.) *Foundations of Knowledge Acquisition: Cognitive Models of Complex Learning*. (pp. 269-307). Hingham, MA: Kluwer Academic Publishers. (Also available in S. Minton (Ed.) *Machine Learning Methods for Planning and Scheduling*. San Mateo, CA: Morgan Kaufmann. In Press.)

Ruby, D., & Kibler, D. (1991). SteppingStone: An empirical and analytical evaluation. *Proceedings of the Ninth National Conference on Artificial Intelligence* (pp. 527-532). Anaheim, CA: AAAI Press.

Russell, S. J., & Grosz, B. N. (1987). A declarative approach to bias in concept learning. *Proceedings of the Sixth National Conference on Artificial Intelligence* (pp. 505-510). Seattle, WA: Morgan Kaufmann.

Utgoff, P. E. (1986). Shift of bias for inductive concept learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.) *Machine Learning: An Artificial Intelligence Approach, Vol. II*. Los Altos, CA: Morgan Kaufmann.